



BAB II

TINJAUAN PUSTAKA

2.1 Teori Umum

2.1.1 Pengertian Perangkat Keras

Menurut Putri et al. (2017:71), “Pengertian dari *Hardware* atau dalam bahasa Indonesia disebut juga dengan nama perangkat keras adalah salah satu komponen dari sebuah komputer yang sifat alatnya bisa dilihat dan diraba secara langsung atau yang berbentuk nyata”.

Sedangkan menurut Stair dikutip Amara (2016:83), “Perangkat keras, yaitu perangkat keras komponen untuk melengkapi kegiatan masukan data, memproses data, dan keluaran data”.

Berdasarkan pendapat para ahli diatas dapat disimpulkan bahwa, perangkat keras adalah suatu alat yang digunakan untuk mengolah sebuah data untuk melengkapi kegiatan masukan data, memproses data, dan keluaran data.

2.1.2 Pengertian Internet

Menurut Zabar dan Novianto (2015:69), “Internet adalah suatu jaringan komputer yang saling terhubung untuk keperluan komunikasi dan informasi”.

Sedangkan menurut Greenlaw dan Hep dikutip Sherlyanita dan Rakhmawati (2016:17), “Internet adalah sebuah sistem informasi global yang terhubung secara logika oleh *address* yang unik secara global yang berbasis pada *Internet Protocol* (IP), mendukung komunikasi dengan menggunakan TCP/IP, menyediakan, menggunakan, dan membuatnya bisa diakses baik secara umum maupun secara khusus”.

Berdasarkan pendapat para ahli diatas dapat disimpulkan bahwa, Internet adalah suatu jaringan global yang saling terhubung oleh *address* yang unik untuk keperluan komunikasi dan informasi dan bisa diakses baik secara umum maupun secara khusus.



2.1.3 Pengertian Perangkat Lunak

Menurut Swara dan Pebriadi (2016:28), “Perangkat Lunak merupakan seluruh perintah yang digunakan untuk memproses informasi. Perangkat lunak dapat berupa program maupun prosedur yang di dalamnya merupakan kumpulan perintah yang dimengerti oleh komputer”.

Sedangkan menurut Maulana (2014:21), “Perangkat Lunak merupakan abstraksi fisik yang memungkinkan kita untuk berbicara dengan mesin perangkat keras.

Berdasarkan pendapat para ahli diatas dapat disimpulkan bahwa, Perangkat Lunak adalah seluruh perintah yang memungkinkan kita untuk berbicara dengan mesin perangkat keras.

2.1.4 Pengertian Sistem

Menurut Sitanggang (2019:63), “Sistem berasal dari bahasa Latin (*systema*) dan bahasa Yunani (*sustema*) adalah sekumpulan unsur atau elemen yang saling berkaitan dan saling mempengaruhi dalam melakukan kegiatan bersama untuk mencapai suatu tujuan”.

Sedangkan menurut Mulyadi dikutip Sitanggang (2019:63), “Sistem adalah suatu jaringan prosedur yang dibuat menurut pola yang terpadu untuk melaksanakan kegiatan pokok perusahaan”.

Berdasarkan pendapat para ahli diatas dapat disimpulkan bahwa, sistem adalah suatu kumpulan atau himpunan yang saling berkaitan untuk melaksanakan kegiatan pokok perusahaan.

2.1.5 Metode Pengembangan Perangkat Lunak

Sukanto dan Shalahudin (2016:28), menjelaskan tentang metode pengembangan sistem yaitu *waterfall*. Metode air terjun (*waterfall*) sering juga disebut model sekuensial linier (*sequential linear*) atau alur hidup klasik (*classic*



life cycle). Model air terjun menyediakan pendekatan alur hidup terurut mulai dari analisis, desain, pengkodean, pengujian dan pemeliharaan.

1. Analisis Kebutuhan Perangkat Lunak

Proses pengumpulan kebutuhan dilakukan secara intensif untuk menspesifikasikan kebutuhan sistem agar dapat dipahami perangkat lunak seperti apa yang dibutuhkan oleh *user*. Spesifikasi kebutuhan perangkat lunak pada tahap ini perlu untuk didokumentasikan.

2. Desain

Desain perangkat lunak adalah proses multi langkah yang fokus pada desain pembuatan program sistem termasuk struktur data, arsitektur perangkat lunak, representasi antarmuka dan prosedur pengkodean.

3. Pembuatan Kode Barang

Pada tahap pengkodean, desain harus ditranslasikan ke dalam program perangkat lunak. Hasil dari tahap ini adalah program komputer sesuai dengan desain yang telah dibuat pada tahap desain.

4. Pengujian

Pengujian fokus pada perangkat lunak secara dari segi logik dan fungsional dan memastikan bahwa semua bagian sudah diuji. Hal ini dilakukan untuk meminimalisir kesalahan (*error*) dan memastikan keluaran yang dihasilkan sesuai dengan yang diinginkan. Dalam penelitian ini pengujian sistem akan menggunakan pengujian Black-Box. Pengujian Black-box berfokus pada persyaratan fungsional perangkat lunak. Dengan demikian, pengujian metode ini memungkinkan perekayasa perangkat lunak mendapatkan serangkaian kondisi input yang sepenuhnya menggunakan semua persyaratan fungsional untuk suatu program.

5. Pendukung (*support*) dan pemeliharaan (*maintenance*)

Tidak menutup kemungkinan sebuah perangkat lunak mengalami perubahan ketika sudah mengirimkan ke *user*. Perubahan bisa terjadi adanya kesalahan yang muncul dan tidak terdeteksi saat pengujian atau perangkat lunak harus beradaptasi dengan lingkungan baru.



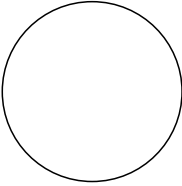
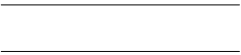
2.2 Teori Khusus

2.2.1 Pengertian DFD (*Data Flow Diagram*)

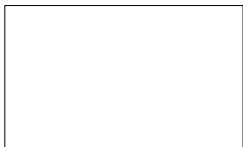

Sukanto dan Shalahuddin (2016:71), *Data Flow Diagram* (DFD) adalah representasi grafik yang menggambarkan aliran informasi dan transformasi informasi yang diaplikasikan sebagai data yang mengalir dari masukan (*input*) dan keluaran (*ouput*).

Adapun notasi-notasi pada DFD (Edward Yourdon dan Tom DeMarco) adalah sebagai berikut :

Tabel 2.1 Simbol-simbol *Data Flow Diagram*

No.	Notasi	Keterangan
1.		Proses atau fungsi atau prosedur, pada pemodelan perangkat lunak yang akan diimplementasikan dengan pemrograman terstruktur, maka pemodelan notasi inilah yang harusnya menjadi fungsi atau prosedur di dalam kode program
2.		File basisdata atau penyimpanan (<i>storage</i>); pada pemodelan perangkat lunak yang akan diimplementasikan dengan pemrograman terstruktur, maka pemodelan notasi inilah yang harusnya dibuat menjadi tabel-tabel basis data yang dibutuhkan, tabel-tabel ini juga harusnya sesuai dengan perancangan tabel-tabel pada basis data (<i>Entity Relationship Diagram</i> (ERD), <i>Conceptual Data Model</i> (CDM), <i>Physical Data Model</i> (PDM)) Catatan :

Lanjutan Tabel 2.1 Simbol-simbol *Data Flow Diagram*

No.	Notasi	Keterangan
		Nama yang diberikan pada sebuah penyimpanan biasanya kata benda
3.		<p>Entitas luar (<i>external entity</i>) atau masukan (<i>input</i>) atau keluaran (<i>output</i>) atau orang yang memakai/berinteraksi dengan perangkat lunak yang dimodelkan atau sistem lain yang terkait dengan aliran data dari sistem yang dimodelkan</p> <p>Catatan :</p> <p>Nama yang digunakan pada masukan (<i>input</i>) atau keluaran (<i>output</i>) biasanya berupa kata benda.</p>
4.		<p>Aliran data: merupakan data yang dikirim antar proses, dari penyimpanan ke proses, atau dari proses masukan (<i>input</i>) atau keluaran (<i>output</i>)</p> <p>Catatan:</p> <p>Nama yang digunakan pada aliran data biasanya berupa kata benda, dapat diawali dengan kata data misalnya “data siswa” atau tanpa kata data misalnya “siswa”.</p>

(Sumber : Sukamto dan Shalahuddin, 2016:71-72)

Sukamto dan Shalahuddin (2016:72-73) menjelaskan, Berikut ini adalah tahapan-tahapan perancangan dengan menggunakan DFD:

1. Membuat DFD Level 0 atau sering disebut juga Context Diagram



DFD Level 0 menggambarkan sistem yang akan dibuat sebagai suatu entitas tunggal yang berinteraksi dengan orang maupun sistem lain. DFD Level 0 digunakan untuk menggambarkan interaksi antara sistem yang akan dikembangkan dengan entitas luar.

2. Membuat DFD Level 1

DFD Level 1 digunakan untuk menggambarkan modul-modul yang ada dalam sistem yang akan dikembangkan. DFD Level 1 merupakan hasil *breakdown* DFD Level 0 yang sebelumnya sudah dibuat.

3. Membuat DFD Level 2

Modul-modul pada DFD Level 1 dapat di-breakdown menjadi DFD Level 2. Modul mana saja yang harus di-breakdown lebih lebih detail tergantung pada tingkat kedetailan modul tersebut. Apabila modul tersebut sudah cukup detail dan rinci maka modul tersebut sudah tidak perlu untuk di-breakdown lagi. Untuk sebuah sistem, jumlah DFD Level 2 sama dengan jumlah modul ada DFD Level 1 yang di-breakdown.

4. Membuat DFD Level 3 dan seterusnya

DFD Level 3,4,5 dan seterusnya merupakan breakdown dari modul pada DFD diatasnya. Breakdown pada level 3,4,5, dan seterusnya aturannya sama persis dengan DFD Level 1 atau Level 2.

2.2.2 Pengertian *Block Chart*

Kristanto (2008:68), *Block Chart* berfungsi untuk memodelkan masukan, keluaran, proses maupun transaksi dengan menggunakan simbol-simbol tertentu. Pembuatan blockchart harus memudahkan bagi pemakai dalam memahami alur dari sistem atau transaksi.

Adapun simbol-simbol yang sering digunakan dalam *block chart* dapat dilihat pada tabel berikut ini:

Tabel 2.2 Simbol-Simbol *Block Chart*

No.	Simbol	Keterangan
1.		Menandakan dokumen, bisa dalam bentuk surat, formulir, buku/bendel/berkas atau cetakan.
2.		Multi dokumen.
3.		Proses manual.
4.		Proses yang dilakukan oleh komputer.
5.		Menandakan dokumen yang diarsipkan (arsip manual).
6.		Data penyimpanan (<i>data storage</i>).
7.		Terminasi yang mewakili simbol tertentu untuk digunakan pada aliran lain pada halaman yang sama.
8.		Terminasi yang menandakan awal dan akhir dari suatu aliran.
9.		Pengambilan keputusan (<i>decision</i>).
10.		Layar peraga (<i>monitor</i>).
11.		Pemasukan data secara manual.

(Sumber : Kristanto, 2008:68-70)

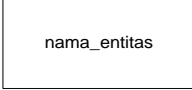
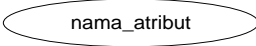
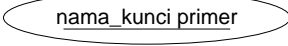


2.2.3 Pengertian ERD (*Entity Relational Diagram*)

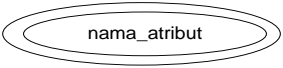
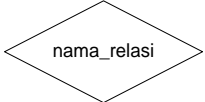
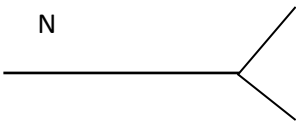
Sukamto dan Shalahuddin (2016:50), *Entity Relational Diagram* (ERD) digunakan untuk pemodelan basis data relasional. ERD memiliki beberapa aliran notasi seperti notasi Chen (dikembangkan oleh Peter Chen), Barker (dikembangkan oleh Richard Barker, Ian Palmer, Harry Ellis), notasi Crow's Foot, dan beberapa notasi lain.

Adapun simbol-simbol yang digunakan pada ERD dengan notasi Chen adalah sebagai berikut:

Tabel 2.3 Simbol-simbol *Entity Relational Diagram*

No.	Simbol	Deskripsi
1.	<p>Entitas / <i>entity</i></p> 	Entitas merupakan data inti yang akan disimpan; bakal tabel pada basis data; benda yang memiliki data dan harus disimoan datanya agar dapat diakses oleh aplikasi komputer; penamaan entitas biasanya lebih ke kata benda dan belum merupakan nama tabel komputer; penamaan entitas biasanya lebih ke kata benda dan belum merupakan nama tabel.
2.	<p>Atribut</p> 	<i>Field</i> atau kolom data yang butuh disimpan dalam suatu entitas.
3.	<p>Atribut kunci primer</p> 	<i>Field</i> atau kolom data yang butuh disimpan dalam suatu entitas dan digunakan sebagai kunci akses <i>record</i> yang diinginkan; biasanya berupa id; kunci primer dapat lebih dari satu kolom, asalkan kombinasi dari beberapa kolom tersebut dapat bersifat unik (berbeda tanpa ada yang sama).

**Lanjutan Tabel 2.3** Simbol-simbol *Entity Relational Diagram*

No.	Simbol	Deskripsi
4.	Atribut multinilai / <i>multivalue</i> 	<i>Field</i> atau kolom data yang butuh disimpan dalam suatu entitas yang dapat memiliki nilai lebih dari satu
5.	Relasi 	Relasi yang menghubungkan antar entitas; biasanya diawali dengan kata kerja.
6.	Asosiasi / <i>association</i> 	Penghubung antar relasi dan entitas di mana di kedua ujungnya memiliki multiplicity kemungkinan jumlah pemakaian Kemungkinan jumlah maksimum keterhubungan antara entitas satu dengan entitas yang lain disebut dengan kardinalitas. Misalkan ada kardinalitas 1 ke N atau sering disebut dengan one to many menghubungkan entitas A dan entitas B maka

(Sumber: Sukanto dan Shalahuddin , 2016:50-51)




2.2.4 Pengertian *Flowchart*

Kadir (2017:39) menjelaskan diagram alir dapat dikelompokkan ke dalam dua kategori, yaitu diagram alir sistem (*system flowchart*) dan diagram alir program (*program flowchart*).


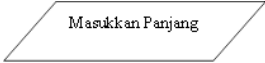
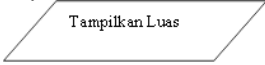
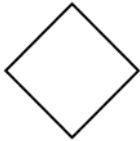
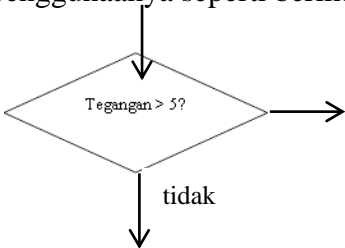
1. Diagram alir sistem biasa digunakan oleh analis sistem untuk menggambarkan aliran data atau struktur *file* di dalam suatu sistem.
2. Diagram alir program biasa dipakai oleh pemrogram untuk menjelaskan langkah-langkah di dalam suatu program.

Adapun simbol-simbol yang digunakan pada diagram alir adalah sebagai berikut:





**Tabel 2.4** Simbol-simbol Diagram Alir (*Flowchart*)

No.	Simbol	Keterangan
1.	 (terminator)	<p>Simbol ini digunakan untuk menyatakan titik awal atau titik akhir diagram alir. Kemungkinan isinya seperti berikut:</p> <div style="display: flex; justify-content: space-around; align-items: center;"> <div style="border: 1px solid black; border-radius: 15px; padding: 5px 20px;">Mulai</div> <div style="border: 1px solid black; border-radius: 15px; padding: 5px 20px;">Selesai</div> </div> <p>Dalam bahasa Inggris, kata <i>Start</i> dan <i>End</i> biasa dipakai.</p>
2.	 (proses)	<p>Simbol ini digunakan untuk menyatakan sebarang proses; misalnya untuk menyatakan suatu operasi aritmetika.</p>
3.	 (proses terdefinisi)	<p>Simbol ini menyatakan prosedur lain yang telah didiagramalirkan pada tempat lain.</p>

Lanjutan Tabel 2.4 Simbol-simbol Diagram Alir (*Flowchart*)

No.	Simbol	Keterangan
4.	 (input/output atau kadang disebut data)	<p>Simbol ini menyatakan operasi pemasukan data atau penampilan data. Contoh:</p> <p>1.  untuk memasukkan data panjang (membaca data panjang dari <i>keyboard</i>)</p> <p>2.  untuk menampilkan data luas</p>
5.	 (keputusan)	<p>Simbol ini digunakan untuk melakukan pengambilan keputusan. Da^{ya} ini, yang ada dalam simbol ini berupa suatu pertanyaan yang jawabannya berupa dua kemungkinan, yaitu “ya” atau “tidak”. Contoh penggunaanya seperti berikut:</p>  <p>Pada contoh tersebut, tanda panah yang menunjuk ke simbol keputusan menyatakan aliran menuju ke keputusan. Kondisi Tegangan > 5? menyatakan</p>

Lanjutan **Tabel 2.4** Simbol-simbol Diagram Alir (*Flowchart*)

No.	Simbol	Keterangan
6.		<p>suatu pertanyaan yang berbunyi, “Apakah isi Tegangan lebih besar daripada 5?” Kalau jawaban pertanyaan tersebut berupa ya, aliran menuju kanan. Sebaliknya, untuk jawaban tidak, maka aliran menuju ke bawah.</p>
7.	 (konektor)	<p>Simbol ini digunakan untuk menghubungkan ke berbagai bagian dalam diagram alir. Contoh:</p>  <p>Bagian yang sebelah kiri menyatakan bahwa aliran dilanjutkan ke bagian yang ditandai dengan . Gambar sebelah kanan menyatakan bagian yang bertanda . Tanda panah menyatakan arah pemrosesan selanjutnya. Namun, simbol ini juga terkadang digunakan pada pertemuan dua aliran.</p>

Sumber : Kadir (2017:40)



2.2.5 Pengertian Kamus Data

Sukamto dan Shalahuddin (2016:73), kamus data adalah kumpulan daftar elemen data yang mengalir pada sistem perangkat lunak sehingga masukan (input) dan keluaran (ouput) dapat dipahami secara umum (memiliki standar cara penulisan). Kamus data memiliki beberapa simbol sebagai berikut :

Tabel 2.5 Simbol-simbol Kamus Data

No.	Simbol	Keterangan
1.	=	Disusun atau terdiri dari
2.	+	Dan
3.	[]	Baik...atau...
4.	{ }n	N kali diulang/bernilai banyak
5.	()	Data opsional
6.	*...*	Batas komentar

(Sumber: Sukamto dan Shalahuddin, 2016:74)

2.2.6 Metode Pengujian Perangkat Lunak

2.2.6.1 Pengertian Metode Pengujian

Sukamto dan Shalahuddin (2018:272) menyatakan, “Pengujian adalah satu set aktifitas yang direncanakan dan sistematis untuk menguji atau mengevaluasi kebenaran yang diinginkan. Aktifitas pengujian terdiri dari satu set atau sekumpulan langkah dimana dapat menempatkan desain kasus uji yang spesifik dan metode pengujian”.

2.2.6.2 Metode Pengujian

Secara umum pola pengujian perangkat lunak adalah sebagai berikut:

1. .Pengujian dimulai dari level komponen hingga integrasi antar komponen menjadi sebuah sistem.
2. Teknik pengujian berbeda-beda sesuai dengan berbagai isi atau unit uji dalam



Dalam waktu yang berbeda-beda pula bergantung pada pengujian pada bagian mana yang dibutuhkan

3. Pengujian dilakukan oleh pengembang perangkat lunak, dan jika untuk proyek besar, pengujian bisa dilakukan oleh tim uji yang tidak terkait dengan tim pengembang perangkat lunak (independent test group (ITG)).
4. Pengujian dan penirkutuan (debugging) merupakan aktivitas yang berbeda tetapi penirkutuan (debugging) harus diakomodasikan pada berbagai strategi pengujian.

2.2.6.3 Black-Box Testing (Pengujian Kotak Hitam)

Sukanto dan Shalahuddin (2018:275), “Black-box testing (pengujian kotak hitam) yaitu menguji perangkat lunak dari segi spesifikasi fungsional tanpa menguji desain dan kode program. Pengujian dimaksudkan untuk mengetahui apakah fungsi-fungsi masukan dan keluaran dari perangkat lunak sesuai dengan spesifikasi yang dibutuhkan”.

Kasus uji yang dibuat untuk melakukan pengujian kotak hitam harus dibuat dengan kasus benar dan kasus salah, misalkan untuk kasus proses login maka kasus uji coba yang dilakukan adalah:

- a. Jika user memasukkan nama pemakai (username) dan kata sandi (password) yang benar.
- b. Jika user memasukkan nama pemakai (username) dan kata sandi (password) yang salah, misalkan nama pemakai benar tapi kata sandi salah, atau sebaliknya atau keduanya salah.

2.3 Teori Judul

2.3.1 Pengertian Aplikasi

Menurut Noor (2016:77), “Aplikasi komputer merupakan suatu perangkat lunak komputer yang memiliki fungsi tertentu sesuai dengan tujuan yang diinginkan oleh si pembuat aplikasi (*progamer*)”.



Sedangkan menurut Siregar dan Permana (2016:83), “Aplikasi adalah program siap pakai yang dibuat untuk melaksanakan suatu fungsi untuk pengguna atau aplikasi lain”.

Berdasarkan pendapat para ahli diatas dapat disimpulkan bahwa, Aplikasi adalah perangkat lunak yang dibuat untuk melaksanakan suatu fungsi yang diinginkan oleh si pembuat.

2.3.2 Pengertian *Proportional*

Iskandar dan Lestari (2016:11), *proporsional* berarti meletakkan sesuatu pada tempatnya, yaitu menempatkan fakta sesuai dengan proporsinya.

Purwosusilo (2017:101), kata *proportional* berarti sesuai dengan proporsi, sebanding, seimbang atau berimbang.

Berdasarkan pendapat para ahli diatas disimpulkan bahwa, *proportional* adalah meletakkan sesuatu sesuai dengan sebanding, seimbang atau berimbang.

2.3.3 Pengertian *Scheduling*

Kusmindari et al. (2019:181), penjadwalan (*scheduling*) adalah pengaturan waktu dari suatu kegiatan operasi.

Silviani (2020:79), penjadwalan (*scheduling*) yaitu menentukan waktu yang diperlukan untuk melaksanakan tindakan untuk mencapai tujuan atau sasaran.

Berdasarkan pendapat para ahli diatas disimpulkan bahwa, *scheduling* adalah penentuan waktu dari suatu kegiatan operasi untuk mencapai tujuan.

2.3.4 Pengertian Lembaga Penyiaran Publik (LPP) TVRI Sumsel

Lembaga Penyiaran Publik (LPP) TVRI Sumatera Selatan adalah stasiun televisi daerah yang didirikan oleh Yayasan Study Group Televisi Palembang yang bekerjasama dengan pihak Pertamina dan TVRI Pusat, Jakarta pada bulan Maret 1972. Diresmikan pada tanggal 31 Januari 1974 sebagai Lembaga Penyiaran Publik



(LPP) TVRI Sumatera Selatan yang menayangkan program-program daerah untuk provinsi Sumatera Selatan dan berita nasional.

2.3.5 Pengertian *Web Mobile*

Pane et al.. (2020:132), *web mobile* adalah *website* yang bisa diakses dan dilihat menggunakan perangkat seperti *smartphone*.

Sedangkan menurut Brian Fling dikutip Abdurahman (2016:51), “*Mobile Websites (Situs Web Mobile)* merupakan situs web yang dirancang khusus untuk perangkat *mobile*. Situs *web mobile* sering memiliki desain yang sederhana dan biasanya bersifat memberikan informasi.

Berdasarkan pendapat para ahli diatas dapat disimpulkan bahwa, *Web Mobile* adalah situs *website* yang bisa diakses melalui perangkat *mobile* dan biasanya bersifat informasi.

2.3.6 Pengertian *Webview*

Suryana (2019:256), “*Webview* adalah tampilan untuk menampilkan konten web dan HTML di aplikasi Anda. Tidak seperti *OpenUri*, yang membawa pengguna ke *browser* web di perangkat, *Webview* menampilkan konten HTML di dalam aplikasi Anda”.

Syahputra (2019:4), “*Webview* merupakan salah satu komponen di *flutter* yang digunakan untuk menampilkan url kedalam bentuk tampilan atau halaman”.

Berdasarkan pendapat para ahli diatas dapat disimpulkan bahwa, *Webview* adalah komponen sistem yang digunakan untuk menampilkan konten HTML di dalam aplikasi.

2.3.7 Pengertian Aplikasi *Proportional Scheduling* Penyiaran Stasiun TVRI Sumsel Berbasis *Mobile Webview*

Aplikasi *Proportional Scheduling* Penyiaran Stasiun TVRI Sumsel Berbasis *Mobile Webview* adalah suatu aplikasi untuk mempermudah pegawai dalam



menyeimbangkan porsi jadwal penyiaran dan produksi Stasiun TVRI Sumsel sesuai dengan target yang telah ditentukan oleh TVRI Pusat.

2.4 Teori Program

2.4.1 Pengertian Basis Data

Menurut Swara dan Pebriadi (2016:30), “Basis data atau *Database* adalah kumpulan informasi yang disusun dan merupakan suatu kesatuan yang utuh yang disimpan di dalam perangkat keras (komputer) secara sistematis sehingga dapat diolah menggunakan perangkat lunak”.

Sedangkan menurut Pahlevi et al. (2018:28), “*Database* adalah sebuah tempat penyimpanan yang besar dimana terdapat kumpulan data yang tidak hanya berisi data operasional tetapi juga deskripsi data”.

Berdasarkan pendapat para ahli di atas dapat disimpulkan bahwa, basis data adalah sebuah tempat penyimpanan yang besar dimana terdapat kumpulan informasi yang disusun dan yang disimpan di dalam perangkat keras sehingga dapat diolah menggunakan perangkat lunak.

2.4.2 Pengertian XAMPP

Menurut Noor (2016:78), “XAMPP Pratama (2016:18), XAMPP adalah *software web server MySQL* yang didukung dengan bahasa pemrograman PHP untuk membuat *website* yang dinamis”.

Sedangkan menurut Laisina et al. (2018:140), “XAMPP adalah perangkat lunak bebas, yang mendukung banyak sistem operasi, merupakan kompilasi dari beberapa program”.

Berdasarkan pendapat para ahli diatas dapat disimpulkan bahwa, XAMPP adalah perangkat lunak bebas yang didukung dengan bahasa pemrograman PHP untuk membuat *website* yang dinamis, merupakan kumpulan dari beberapa program.



2.4.3 Pengertian MySQL

Menurut Pahlevi et al. (2018:28) “MySQL adalah salah satu jenis *database* yang banyak digunakan untuk membuat aplikasi berbasis web yang dinamis”.

Sedangkan menurut Noor (2016:77), “MySQL adalah salah satu jenis *database server* yang sangat terkenal.”

Berdasarkan pendapat para ahli diatas dapat disimpulkan bahwa, MySQL adalah salah satu jenis *database server* yang sangat terkenal dan banyak digunakan untuk membuat aplikasi berbasis web yang dinamis.

2.4.4 Pengertian HTML

Menurut Pahlevi et al. (2018:28), “HTML (*Hyper Text Mark Up Language*) merupakan bahasa yang digunakan untuk mendeskripsian struktur sebuah halaman web”.

Sedangkan menurut Harison dan Syarif (2016:43) “*HyperText Markup Language* (HTML) adalah sebuah bahasa *markup* yang digunakan untuk membuat sebuah halaman web, menampilkan berbagai informasi di dalam sebuah penjelajah web internet dan *forming hypertext* sederhana yang ditulis kedalam berkas format ASCII agar dapat menghasilkan tampilan wujud yang terintegrasi.

Berdasarkan pendapat para ahli diatas dapat disimpulkan bahwa, HTML merupakan bahasa yang digunakan untuk mendeskripsikann struktur halaman web, menampilkan berbagai informasi, dan *forming hypertext* sederhana.

2.4.5 Pengertian PHP

Menurut Harison dan Syarif (2016:42), “PHP adalah sebuah bahasa pemrograman yang berjalan dalam sebuah *web-server (serverside)*”.

Sedangkan menurut Laisina et al. (2018:141), “PHP adalah bahasa perograman yang digunakan secara luas untuk penanganan pembuatan dan pengembangan sebuah web dan bisa digunakan pada HTML”.



Berdasarkan pendapat para ahli diatas dapat disimpulkan bahwa, sebuah bahasa pemrograman yang berjalan dalam sebuah *web-server* yang digunakan secara luas untuk penanganan pembuatan dan pengembangan sebuah web dan bisa digunakan pada HTML .



Gambar 2.1. *Tampilan Logo PHP*

2.4.6 Pengertian JavaScript

Menurut Pahlevi et al. (2018:28), “*JavaScript* adalah bahasa pemrograman web yang bersifat *Client Side Programming Language*. *Client Side Programming Language* adalah tipe bahasa pemrograman yang pemrosesannya dilakukan oleh *client*. Aplikasi *client* yang dimaksud merujuk kepada *web browser* seperti Google Chrome, Mozilla Firefox, Opera Mini dan sebagainya”.

2.4.7 Pengertian CSS (*Cascading Style Sheet*)

Menurut Pahlevi et al. (2018:28), “CSS kepanjangan dari *Cascading Style Sheet* adalah bahasa-bahasa yang merepresentasikan halaman web. Seperti warna, *layout*, dan *font*”.

Sedangkan menurut Noor (2016:78), “CSS adalah kependekan dari *Cascading Style Sheet*. CSS merupakan salah satu kode pemrograman yang bertujuan untuk menghias dan mengatur gaya tampilan/*layout* halaman web supaya lebih elegan dan menarik”.

Kesimpulannya adalah CSS merupakan bahasa-bahasa yang merepresentasikan halaman web seperti warna, *layout*, dan *font*, yang bertujuan untuk menghias dan mengatur gaya tampilan/*layout* halaman web supaya lebih elegan dan menarik.



2.4.8 Pengertian Notepad++

Menurut Angga Reza Palevi1 dan Krisnawati dikutip Siregar et al. (2018:115), “*Notepad++* adalah sebuah aplikasi *text editor* yang bersifat gratis. *Notepad* menitik beratkan kegunaan aplikasi untuk *editing text* dalam waktu yang cepat dan praktis. *Notepad++* mendukung banyak format bahasa pemrograman seperti PHP, HTML, *Java Script* dan CSS. Aplikasi ini dapat diunduh secara bebas di alamat “.

Sedangkan menurut Poipessy dan Umasangadji (2018:40), “*Notepad++* adalah *code editor* (*software* penyunting kode) yang mendukung berbagai bahasa pemrograman seperti HTML, CSS, PHP, JAVA, dan lain-lain yang dapat bekerja pada Sistem Operasi Windows. *Kelebihan Notepad++* jika dibanding *notepad* bawaan *windows* adalah memiliki kelengkapan fitur untuk mempermudah pengguna saat mengedit kode termasuk saat mengedit HTML dan kode CSS”.

Berdasarkan pendapat para ahli diatas dapat disimpulkan bahwa, *Notepad++* merupakan sebuah aplikasi *text editor* yang mendukung berbagai bahasa pemrograman seperti HTML, CSS, PHP, JAVA, dan lain-lain, memiliki kelengkapan fitur untuk mempermudah pengguna saat mengedit kode.